DOI:
10.1002/nla.2155

Document Version
Peer reviewed version

NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 2018; **00**:1–21
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nla

Convergence of the multigrid reduction in time algorithm for the linear elasticity equations

A. Hessenthaler[1]*, D. Nordsletten[2], O. Röhrle[1], J. B. Schroder[3], R. D. Falgout[3]

[1]*Institute of Applied Mechanics (CE), University of Stuttgart, Pfaffenwaldring 7, 70569 Stuttgart, Germany*
[2]*Division of Imaging Sciences and Biomedical Engineering, King's College London, 4th Floor, Lambeth Wing St. Thomas Hospital London, SE1 7EH, UK*
[3]*Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-561, Livermore, CA 94551*

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52–07NA27344, LLNL-JRNL-731168.

SUMMARY

This paper presents some recent advances for parallel-in-time methods applied to linear elasticity. With recent computer architecture changes leading to stagnant clock speeds, but ever increasing numbers of cores, future speedups will be available through increased concurrency. Thus, sequential algorithms, such as time stepping, will suffer a bottleneck. This paper explores multigrid reduction in time (MGRIT) for an important application area, linear elasticity. Previously, efforts at parallel-in-time for elasticity have experienced difficulties, for example, the beating phenomenon. As a result, practical parallel-in-time algorithms for this application area currently do not exist. This paper proposes some solutions made possible by MGRIT (*e.g.*, slow temporal coarsening and FCF-relaxation) and more importantly, a different formulation of the problem that is more amenable to parallel-in-time methods. Using a recently developed convergence theory for MGRIT and Parareal, we show that the changed formulation of the problem avoids the instability issues and allows reduction of the error using two temporal grids. We then extend our approach to the multilevel case, where we demonstrate how slow temporal coarsening improves convergence. The paper ends with supporting numerical results showing a practical algorithm enjoying speedup benefits over the sequential algorithm. Copyright © 2018 John Wiley & Sons, Ltd.

Received . . .

KEY WORDS: multigrid reduction in time (MGRIT); parallel-in-time; linear elasticity; convergence estimate

1. INTRODUCTION

Clock rates of recent computer architectures have become stagnant, while the available number of parallel processors has increased and continues to increase rapidly. Thus, simulation algorithms need to allow greater concurrency to exploit massively parallel hardware and further reduce wall clock time. One severe sequential bottleneck in many parallel application codes is the use of sequential time integration methods. This sequential bottleneck limits parallelism to the spatial component of a space-time problem.

One such research area where sequential time-stepping limits performance is fluid-structure interaction (FSI) research. Here, the system under consideration models the interaction between

*Correspondence to: A. Hessenthaler, Institute of Applied Mechanics (CE), University of Stuttgart, Pfaffenwaldring 7, 70569 Stuttgart, Germany. Email: hessenthaler@mechbau.uni-stuttgart.de

fluids and (often deformable) solid structures or particles, for example, in biomedical or aerospace engineering applications [1, 2, 3]. Well-established parallelization techniques, such as spatial domain decomposition methods, provide a straightforward and scalable approach for reducing the wall clock time for many FSI algorithms. However, spatial parallelism saturates when communication tasks become dominant over computation tasks. This often prohibits use of parallel resources beyond an optimal processor count, leaving large numbers of processors unused. On the other hand, parallel-in-time integration methods provide means of introducing an additional layer of parallelism with the potential to greatly enhance efficient and more exhaustive usage of parallel resources. Despite its long history [4, 5], parallel-in-time methods are rarely used in the large-scale simulation community. Only a few efforts exist that use parallel-in-time ideas in real-world simulations, including reservoir simulation [6], fusion research [7] and numerical weather prediction [8].

In the field of FSI research, barriers to using parallel-in-time methods are largely due to an instability arising in the parallel-in-time integration of one of its subproblems, namely, dynamic structural mechanics. The instability is known as the beating phenomenon [9, 10]. Although this instability can be overcome by filtering the natural modes of a given structural dynamics problem [10, 11], only small scale parallelism has been explored. This can mainly be related to a relatively expensive projection step and the restriction of the method to two time grid levels. Thus, it is important to obtain a stable and robust parallel-in-time technique that can produce greater parallelism while avoiding the previously mentioned instability. Such a method can then be the foundation of a fast and efficient parallel-in-time method applicable to FSI applications.

The focus of this work is the investigation of the convergence of the multigrid reduction in time (MGRIT) method for the second-order partial differential equation governing the dynamic linear-elastic response of an incompressible solid structure. In a special case (two temporal grids with F-relaxation), MGRIT is equivalent [12] to Parareal [17] but has several benefits. For example, MGRIT allows use of FC-relaxation, slow temporal coarsening and is a non-intrusive algorithm. Moreover, MGRIT is a true multilevel algorithm.

In this paper, we will make use of the recent work of the authors of [12] to use the presented two-grid convergence theory as a design tool for convergent algorithms. One of the advantages of the two-level theory is its applicability as an *a priori* convergence bound. It solely requires the fine-grid and coarse-grid time stepping matrices, the temporal coarsening factor and the number of time steps as inputs and provides a sharp upper bound to the error contraction at each iteration. Additionally, it considers the effects of FC-relaxation. Compared to the semi-algebraic mode analysis (SAMA) presented in [18], it does not diagonalize the two-grid error propagator with Fourier modes but assumes that the fine-grid and coarse-grid time stepping matrices can be diagonalized by the same set of eigenvectors. This assumption, taken together with the fact that MGRIT is a reduction-based method, allows the presented analysis to be performed only on the coarse-grid (see [12]). However, the SAMA analysis would allow for the derivation of a similar convergence bound. See for instance [12], where the analysis technique used here yields equivalent estimates for the heat equation when compared to SAMA.

The beating phenomenon was described qualitatively in [9, 10]. Here, we will show that it is also founded in the two-grid theory. Furthermore, the same analysis provides a powerful tool for estimating convergence *a priori*, leading to a convergent two-grid MGRIT algorithm for the dynamic second-order elasticity equations and a convergent multilevel MGRIT algorithm.

In Section 2, two different backward Euler time discretization schemes are presented (referred to as Scheme I and Scheme II) for a finite element implementation of the second-order hyperbolic elasticity equations. The two schemes are embedded in the MGRIT algorithm, where we provide a description of the parallel-in-time algorithm and discuss the application of the convergence theory presented in [12]. We discuss the difference between both time discretization schemes in Section 3.2. In Section 3, we relate the theoretical analysis of the two-grid convergence to observations in numerical experiments. Further, we demonstrate how to derive a convergent MGRIT algorithm that does not exhibit the instabilities reported in previous work [9, 10] by using [12] as an algorithm design tool. We then extend the numerical experiments to a multilevel case with FMG-cycles in

DOI: 10.1002/nla

Section 3.4, where we consider different numbers of V-cycles per FMG-level. The section concludes with timing results, demonstrating a speedup of up to 5.25 on 192 processors and emphasizing the potential for parallel speedup when using MGRIT for linear elasticity problems.

2. METHODOLOGY

2.1. Notation

Consider the domain $\Omega = \Omega(t) \subset \mathbb{R}^d \times [0, T]$ with Dirichlet boundary Γ^D in d spatial dimensions. Then, $\boldsymbol{X} \in \Omega(0)$ and $\boldsymbol{x} \in \Omega$ denote the reference and current position of a material point and $\nabla_{\boldsymbol{X}}$ and $\nabla_{\boldsymbol{x}}$ denote the respective Lagrangian and Eulerian gradient operators. Further, we define the deformation gradient $\boldsymbol{F} = \nabla_{\boldsymbol{X}} \boldsymbol{x} = \nabla_{\boldsymbol{X}} \boldsymbol{u} + \boldsymbol{I}$ where $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{x}, t) = \boldsymbol{x} - \boldsymbol{X}$ is the displacement of a material point with respect to its position in the reference configuration. The partial derivative operator with respect to time is denoted as ∂_t and $\partial_{tt} = \partial_t \partial_t$.

2.2. Governing equations

Consider the governing equations for the dynamic and linear-elastic response of an incompressible solid structure with given initial data and Dirichlet boundary condition data,

$$\rho \partial_{tt} \boldsymbol{u} - \nabla_{\boldsymbol{x}} \cdot \boldsymbol{\sigma} = \boldsymbol{0} \qquad \text{in } \Omega, \tag{1}$$

$$\nabla_{\boldsymbol{x}} \cdot \partial_t \boldsymbol{u} = 0 \qquad \text{in } \Omega, \tag{2}$$

$$\boldsymbol{u}(\cdot, t) = \boldsymbol{0} \qquad \text{on } \Gamma^D, \tag{3}$$

$$\boldsymbol{u}(\cdot, 0) = \boldsymbol{0}, \quad \boldsymbol{v}(\cdot, 0) = \hat{\boldsymbol{v}}_0 \qquad \text{in } \Omega(0), \tag{4}$$

with density ρ, Cauchy stress tensor $\boldsymbol{\sigma}(\boldsymbol{u}, p) = \mu(\boldsymbol{F} - \boldsymbol{I}) - p\boldsymbol{I}$, material stiffness parameter μ, the hydrostatic pressure variable p and initial velocity $\hat{\boldsymbol{v}}_0$.

Equation 1 can be transformed to a system of first-order equations,

$$\partial_t \boldsymbol{u} = \boldsymbol{v} \qquad \text{in } \Omega, \tag{5}$$

$$\rho \partial_t \boldsymbol{v} = \nabla_{\boldsymbol{x}} \cdot \boldsymbol{\sigma} \qquad \text{in } \Omega, \tag{6}$$

$$\nabla_{\boldsymbol{x}} \cdot \boldsymbol{v} = 0 \qquad \text{in } \Omega, \tag{7}$$

$$\boldsymbol{u}(\cdot, t) = \boldsymbol{0} \qquad \text{on } \Gamma^D, \tag{8}$$

$$\boldsymbol{u}(\cdot, 0) = \boldsymbol{0}, \quad \boldsymbol{v}(\cdot, 0) = \hat{\boldsymbol{v}}_0 \qquad \text{in } \Omega(0), \tag{9}$$

with velocity $\boldsymbol{v}$.

To reduce the complexity of the computational model, Equation 5 is eliminated from the system of equations in the following by including it implicitly. That is, we solve for velocity $\boldsymbol{v}$ and hydrostatic pressure p and update the displacement variable based on the solution for the velocity variable. Further, we note that all quantities are computed on the reference domain $\Omega(0)$. That is, linear-elastic response is assumed and higher-order effects of the deforming domain are neglected.

2.3. Time discretization

We decompose the temporal domain [0, T] by $N_t + 1$ equidistant time points, such that

$$t_i = i \cdot \delta_{N_t}, \quad \text{with } i = 0, \ldots, N_t, \tag{10}$$

with time step size $\delta_{N_t} = T/N_t$, initial time $t_0 = 0$ and final time $t_{N_t} = T$. Then, velocity, pressure and displacement at time point t_i are denoted as $[\boldsymbol{v}_i, p_i, \boldsymbol{u}_i]^T = [\boldsymbol{v}(\cdot, t_i), p(\cdot, t_i), \boldsymbol{u}(\cdot, t_i)]^T$. We now introduce the two discretization schemes (Scheme I and Scheme II) examined here.

Scheme I is considered as the default scheme in our application code CHeart [13] and is motivated by better conserving the energy in the system for large time step sizes (see Section 3.2; for more details, see [14, 15, 16]). On the other hand, Scheme II is proposed as an improvement for parallel-in-time methods with the capability of predicting amplitudes of oscillation with comparable quality for practical time step sizes.

DOI: 10.1002/nla

2.3.1. Scheme I

We approximate the partial derivative operator ∂_t in Equation 5 using the midpoint rule,

$$\boldsymbol{u}_i = \boldsymbol{u}_{i-1} + \delta_{N_t} \frac{\boldsymbol{v}_i + \boldsymbol{v}_{i-1}}{2} \qquad \text{in } \Omega_0. \tag{11}$$

The partial derivative operator in Equation 6 is discretized using the backward Euler scheme where we substitute with Equation 11. Thus, we search $(\boldsymbol{v}_i, p_i)$ for all $i = 1, \dots, N_t$, such that,

$$\rho \boldsymbol{v}_i - \frac{\mu \delta_{N_t}^2}{2} \nabla_{\boldsymbol{X}}^2 \boldsymbol{v}_i + \delta_{N_t} \nabla_{\boldsymbol{X}} p_i = \rho \boldsymbol{v}_{i-1} + \frac{\mu \delta_{N_t}^2}{2} \nabla_{\boldsymbol{X}}^2 \boldsymbol{v}_{i-1} + \mu \delta_{N_t} \nabla_{\boldsymbol{X}}^2 \boldsymbol{u}_{i-1} \qquad \text{in } \Omega_0, \tag{12}$$

$$\nabla_{\boldsymbol{X}} \cdot \boldsymbol{v}_i = -\nabla_{\boldsymbol{X}} \cdot \boldsymbol{v}_{i-1} \qquad \text{in } \Omega_0, \tag{13}$$

and compute the displacement $\boldsymbol{u}_i$ according to Equation 11 after each solve.

2.3.2. Scheme II

We can obtain a slightly different scheme by approximating the partial derivative operator in Equation 5 as,

$$\boldsymbol{u}_i = \boldsymbol{u}_{i-1} + \delta_{N_t} \boldsymbol{v}_i \qquad \text{in } \Omega_0. \tag{14}$$

Thus, in Scheme II we search the time-discrete $(\boldsymbol{v}_i, p_i)$ for all $i = 1, \dots, N_t$, such that,

$$\rho \boldsymbol{v}_i - \mu \delta_{N_t}^2 \nabla_{\boldsymbol{X}}^2 \boldsymbol{v}_i + \delta_{N_t} \nabla_{\boldsymbol{X}} p_i = \rho \boldsymbol{v}_{i-1} + \mu \delta_{N_t} \nabla_{\boldsymbol{X}}^2 \boldsymbol{u}_{i-1} \qquad \text{in } \Omega_0, \tag{15}$$

$$\nabla_{\boldsymbol{X}} \cdot \boldsymbol{v}_i = 0 \qquad \text{in } \Omega_0, \tag{16}$$

The modification in Equation 14 is motivated by the observation that Scheme I does not yield a convergent MGRIT algorithm and exhibits the same instability as described in previous work [9, 10] (see Section 3.3). Scheme II does not suffer from this instability.

2.4. Space discretization

The domain Ω_i was discretized using quadrilateral elements, Ω_i^h. Finite element discretizations were constructed using inf-sup stable $\mathbb{Q}^2 - \mathbb{Q}^1$ Taylor-Hood elements for velocity and pressure, $\boldsymbol{v}_i^h$ and p_i^h, and $\mathbb{Q}^2$ elements for displacement, $\boldsymbol{u}_i^h$. The superscript h denotes the space-discretized version of the domain and state variables. As we do not consider spatial refinement or coarsening, we omit the superscript h for the remainder of this work.

2.4.1. Scheme I

Equation 12 and Equation 13 are discretized in space, which leads to the following problem: find the space-time discrete solution vector $[\boldsymbol{v}_i, \boldsymbol{p}_i]^T$, such that for each $i = 1, \dots, N_t$,

$$(\rho \boldsymbol{M} - \frac{\mu \delta_{N_t}^2}{2} \boldsymbol{K}) \boldsymbol{v}_i + \delta_{N_t} \boldsymbol{B}^T \boldsymbol{p}_i = (\rho \boldsymbol{M} + \frac{\mu \delta_{N_t}^2}{2} \boldsymbol{K}) \boldsymbol{v}_{i-1} + \mu \delta_{N_t} \boldsymbol{K} \boldsymbol{u}_{i-1} \qquad \text{in } \Omega_0, \tag{17}$$

$$\boldsymbol{B} \boldsymbol{v}_i = -\boldsymbol{B} \boldsymbol{v}_{i-1} \qquad \text{in } \Omega_0, \tag{18}$$

where $\boldsymbol{M}$ is the mass matrix and $\boldsymbol{K}$ and $\boldsymbol{B}$ refer to the discretized weak form Laplacian and divergence operators $\nabla_{\boldsymbol{X}}^2()$ and $\nabla_{\boldsymbol{X}} \cdot ()$. Note, after solving for a given $[\boldsymbol{v}_i, \boldsymbol{p}_i]^T$, we can update the displacement $\boldsymbol{u}_i$ from Equation 11. Writing Equation 17 and Equation 18 in matrix form and including the update given in Equation 11 yields the following linear system,

$$\begin{bmatrix} \rho \boldsymbol{M} - \frac{\mu \delta_{N_t}^2}{2} \boldsymbol{K} & \delta_{N_t} \boldsymbol{B}^T & \boldsymbol{0} \\ \boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} \\ -\frac{\delta_{N_t}}{2} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_i \\ \boldsymbol{p}_i \\ \boldsymbol{u}_i \end{bmatrix} = \begin{bmatrix} \rho \boldsymbol{M} + \frac{\mu \delta_{N_t}^2}{2} \boldsymbol{K} & \boldsymbol{0} & \mu \delta_{N_t} \boldsymbol{K} \\ -\boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} \\ \frac{1}{2} \delta_{N_t} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_{i-1} \\ \boldsymbol{p}_{i-1} \\ \boldsymbol{u}_{i-1} \end{bmatrix}. \tag{19}$$

DOI: 10.1002/nla

Denoting the linear operators on the left and right hand sides by,

$$\boldsymbol{D}^I := \begin{bmatrix} \rho\boldsymbol{M} - \frac{\mu\delta_{N_t}^2}{2}\boldsymbol{K} & \delta_{N_t}\boldsymbol{B}^T & \boldsymbol{0} \\ \boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} \\ -\frac{\delta_{N_t}}{2}\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \quad \boldsymbol{C}^I := \begin{bmatrix} \rho\boldsymbol{M} + \frac{\mu\delta_{N_t}^2}{2}\boldsymbol{K} & \boldsymbol{0} & \mu\delta_{N_t}\boldsymbol{K} \\ -\boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} \\ \frac{1}{2}\delta_{N_t}\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \tag{20}$$

with $\boldsymbol{\Phi}^I := [\boldsymbol{D}^I]^{-1}\boldsymbol{C}^I$ and $\boldsymbol{s}_0^I = \hat{\boldsymbol{s}}_0 := [\hat{\boldsymbol{v}}_0, \boldsymbol{0}, \boldsymbol{0}]^T$, the state variables $\boldsymbol{s}_i^I := [\boldsymbol{v}_i, \boldsymbol{p}_i, \boldsymbol{u}_i]^T$ can be computed by the following equation,

$$\boldsymbol{s}_i^I = \boldsymbol{\Phi}^I \boldsymbol{s}_{i-1}^I \qquad \text{for } i = 1, \ldots, N_t. \tag{21}$$

Note, that for the considered model (linear-elastic, incompressible) and for fixed spatial resolution, the operator $\boldsymbol{\Phi}^I$ is only dependent on time step size δ_{N_t}. That is, the operator $\boldsymbol{\Phi}^I$ only needs to be computed once per time step size.

2.4.2. *Scheme II*

In a similar way, Scheme II can be written as: for each $i = 1, \ldots, N_t$, we seek the space-time discrete $[\boldsymbol{v}_i, \boldsymbol{p}_i]^T$, such that,

$$(\rho\boldsymbol{M} - \mu\delta_{N_t}^2\boldsymbol{K})\boldsymbol{v}_i + \delta_{N_t}\boldsymbol{B}^T\boldsymbol{p}_i = \rho\boldsymbol{M}\boldsymbol{v}_{i-1} + \mu\delta_{N_t}\boldsymbol{K}\boldsymbol{u}_{i-1} \qquad \text{on } \Omega_0, \tag{22}$$

$$\boldsymbol{B}\boldsymbol{v}_i = 0 \qquad \text{on } \Omega_0, \tag{23}$$

and update the displacement $\boldsymbol{u}_i$ from Equation 14 after each solve. In matrix notation, we can write,

$$\begin{bmatrix} \rho\boldsymbol{M} - \mu\delta_{N_t}^2\boldsymbol{K} & \delta_{N_t}\boldsymbol{B}^T & \boldsymbol{0} \\ \boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} \\ -\delta_{N_t}\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_i \\ \boldsymbol{p}_i \\ \boldsymbol{u}_i \end{bmatrix} = \begin{bmatrix} \rho\boldsymbol{M} & \boldsymbol{0} & \mu\delta_{N_t}\boldsymbol{K} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_{i-1} \\ \boldsymbol{p}_{i-1} \\ \boldsymbol{u}_{i-1} \end{bmatrix}. \tag{24}$$

With

$$\boldsymbol{D}^{II} := \begin{bmatrix} \rho\boldsymbol{M} - \mu\delta_{N_t}^2\boldsymbol{K} & \delta_{N_t}\boldsymbol{B}^T & \boldsymbol{0} \\ \boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} \\ -\delta_{N_t}\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \quad \text{and} \quad \boldsymbol{C}^{II} := \begin{bmatrix} \rho\boldsymbol{M} & \boldsymbol{0} & \mu\delta_{N_t}\boldsymbol{K} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}, \tag{25}$$

Equation 24 can be written as,

$$\boldsymbol{s}_i^{II} = \boldsymbol{\Phi}^{II} \boldsymbol{s}_{i-1}^{II} \qquad \text{for } i = 1, \ldots, N_t, \tag{26}$$

where $\boldsymbol{\Phi}^{II} := [\boldsymbol{D}^{II}]^{-1}\boldsymbol{C}^{II}$ and $\boldsymbol{s}_i^{II} := [\boldsymbol{v}_i, \boldsymbol{p}_i, \boldsymbol{u}_i]^T$ with $\boldsymbol{s}_0^{II} = \hat{\boldsymbol{s}}_0 := [\hat{\boldsymbol{v}}_0, 0, \boldsymbol{0}]^T$. Again, the operator $\boldsymbol{\Phi}^{II}$ only depends on time step size δ_{N_t}.

2.5. *Multigrid reduction in time (MGRIT) algorithm*

Based on Equation 21 and Equation 26, the global space-time problem can be written in the linear form,

$$\boldsymbol{A}^S\boldsymbol{s}^S = \begin{bmatrix} \boldsymbol{I} & & & & \\ -\boldsymbol{\Phi}^S & \boldsymbol{I} & & & \\ & -\boldsymbol{\Phi}^S & \boldsymbol{I} & & \\ & & \ddots & \ddots & \\ & & & -\boldsymbol{\Phi}^S & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{s}_0^S \\ \boldsymbol{s}_1^S \\ \boldsymbol{s}_2^S \\ \vdots \\ \boldsymbol{s}_{N_t}^S \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{s}}_0 \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \end{bmatrix} = \hat{\boldsymbol{s}}, \tag{27}$$

with $S \in \{I, II\}$. A traditional time stepping method would solve Equation 27 in a block-forward fashion, whereas the MGRIT algorithm solves Equation 27 iteratively. Both algorithms are $O(N_t)$, however, the constant is bigger for MGRIT [20, 12]. On the other hand, MGRIT enables parallelism in the temporal domain in contrast to sequential time stepping. The parallelism is achieved by

DOI: 10.1002/nla

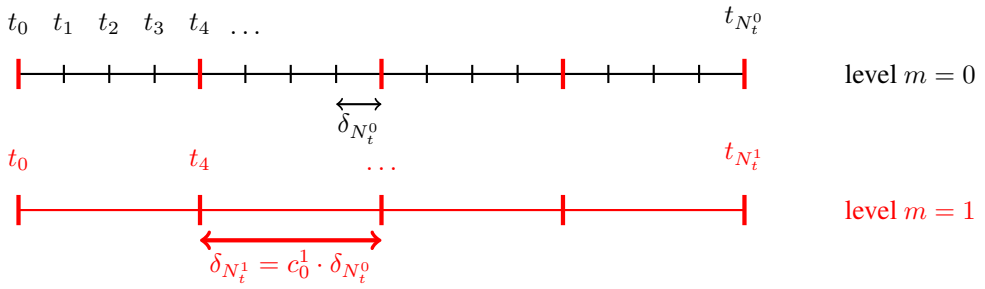


Figure 1. Example of an m_l-level multigrid hierarchy with $m_l = 2$ time grid levels, for time points t_i, time step sizes $\delta_{N_t^m}$ and coarsening factors c_{m-1}^m.

introducing a time grid hierarchy and applying multigrid techniques to the temporal domain. The coarser time grids provide error corrections to the finest time grid, thus accelerating convergence to the solution, while a relaxation process on each time grid reduces the error that cannot be adequately reduced on coarser grids. These two complementary processes, relaxation and coarse-grid correction, form the core of multigrid methods.

Figure 1 illustrates an example of an m_l-level time grid hierarchy with respective time points, time step sizes and coarsening factors. Here, the maximum number of time grid levels is given as $m_l = 2$ (with levels $m = 0, \dots, m_l - 1$). Each level with $m < m_l - 1$ is composed of F- and C-points. F-points only exist on level m and the C-points compose the coarser grid $m + 1$. The coarsening factor c_{m-1}^m (for $m = 1, \dots, m_l - 1$) defines a coarsening of a given level $m - 1$, such that the time step size on level m is $\delta_{N_t^m} = c_{m-1}^m \cdot \delta_{N_t^{m-1}}$ (for $m = 1, \dots, m_l - 1$), with fine grid time step size $\delta_{N_t^0}$ and the coarsest grid's time step size $\delta_{N_t^{m_l-1}}$. Similarly, $c_0^{m_l-1}$ denotes the total coarsening from level 0 to level m_l. Further, F-relaxation means an update of the F-points based on the previous C-points and C-relaxation refers to an update of a C-point based on the previous F-point. For more details, see [20].

For the remainder of this work, we distinguish between the numerical space-time solution obtained with Scheme I with sequential time stepping with a subscript s (sequential) and those quantities obtained from using MGRIT with a subscript p (parallel). For example, the solution vector $\boldsymbol{s}^I$ obtained from sequential time stepping is denoted as $\boldsymbol{s}_s^I$, whereas the solution vector obtained with MGRIT is denoted as $\boldsymbol{s}_p^I$. Similarly, we identify the numerical solution obtained with Scheme II by using the notation $\boldsymbol{s}_s^{II}$ and $\boldsymbol{s}_p^{II}$.

Further, we consider two-grid algorithms with various numbers of FC-relaxation steps and multilevel algorithms using full multigrid (FMG) cycles (*i.e.*, F-cycles). Unless noted otherwise, we either use the one-step integrator $\boldsymbol{\Phi}_{\delta_t^0}^I$ on the fine grid and its coarse grid versions $\boldsymbol{\Phi}_{\delta_t^m}^I$ on grid level m or similarly, $\boldsymbol{\Phi}_{\delta_t^0}^{II}$ on the fine grid and its coarse grid versions $\boldsymbol{\Phi}_{\delta_t^m}^{II}$ on grid level m. The subscript δ_t^m denotes $\delta_{N_t^m}$.

2.5.1. *Two-level algorithm*

Firstly, a two-grid version of the MGRIT algorithm is considered. In this version, r FC-relaxations ($r = 0, 1, 2$) are applied on the fine grid for Scheme I and Scheme II. For $r = 0$, MGRIT is equivalent to Parareal [17]. To restrict the variables from the fine to the coarse grid, a restriction operator $R(\cdot)$ is chosen that purely injects the respective fine grid quantities to the coarse grid. To transfer the coarse grid quantities to the fine grid, an ideal interpolation is employed [20]. The pseudo-code of the linear two-grid version of the MGRIT algorithm is described in Algorithm 1.[†] The coarse-grid operator $\boldsymbol{A}_\Delta^S$ is equivalent to that defined in Equation 27, except that it uses the coarse-grid time-stepper $\boldsymbol{\Phi}_{\delta_t^{m+1}}^S$ and is defined over the smaller coarse time-grid depicted in Figure 1.

[†]The nonlinear version of MGRIT is used in practice and is equivalent to the linear version for linear problems [19].

DOI: 10.1002/nla

The two-grid algorithm with F- and FCF- relaxation will be the basis for predicting and analyzing convergence of the proposed algorithm and for comparing the observed convergence factors with the theoretical upper bounds derived from [12]. The presented analysis will then prove and emphasize the relevance of [12] for the design of convergent parallel-in-time algorithms and motivate the selection of Scheme II to be included in a multilevel hierarchy.

Algorithm 1 Pseudo-code for MGRIT algorithm with V-cycles and $S \in \{I, II\}$, adapted from [12].

1: **repeat**
2: Relax on $\boldsymbol{A}^S \boldsymbol{s}^S = \hat{\boldsymbol{s}}$ (r times FC-relaxation, one F-relaxation) using $\boldsymbol{\Phi}^S_{\delta_t^m}$. ▷ In parallel
3: Compute coarse-grid residual $\boldsymbol{r}_\Delta = R(\hat{\boldsymbol{s}} - \boldsymbol{A}^S \boldsymbol{s}^S)$.
4: Solve the coarse grid correction problem $\boldsymbol{A}^S_\Delta \boldsymbol{e}^S_\Delta = \boldsymbol{r}_\Delta$ using $\boldsymbol{\Phi}^S_{\delta_t^{m+1}}$. ▷ Apply recursively
5: Correct the solution at the fine-grid C-points with $\boldsymbol{e}^S_\Delta$. ▷ In parallel
6: **until** norm of residual is small enough
7: Update the solution at the F-points with $\boldsymbol{\Phi}^S_{\delta_t^0}$. ▷ In parallel

2.5.2. *Multilevel algorithm using FMG-cycles*

In addition to the two-level algorithms, we consider m_l-grid algorithms ($m_l > 2$) with FMG-cycles. A FMG-cycle is achieved by applying Step 4 of Algorithm 1 recursively and performing (at least) one V-cycle as a post-relaxation step at each level [21].

2.5.3. *Theoretical two-grid reduction of the MGRIT residual norm per iteration in the two-grid case*

The theoretical two-grid reduction rate estimates of the global space-time error per MGRIT iteration are based on the two-grid cases discussed in [12]. Firstly, we note that $\boldsymbol{\Phi}^S_{\delta_t^0}$ (and the coarse-grid version $\boldsymbol{\Phi}^S_{\delta_t^1}$) have the sparsity pattern,

$$\begin{bmatrix} [\boldsymbol{\Phi}^S_{\delta_t^m}]_{11} & [\boldsymbol{\Phi}^S_{\delta_t^m}]_{12} & [\boldsymbol{\Phi}^S_{\delta_t^m}]_{13} \\ [\boldsymbol{\Phi}^S_{\delta_t^m}]_{21} & [\boldsymbol{\Phi}^S_{\delta_t^m}]_{22} & [\boldsymbol{\Phi}^S_{\delta_t^m}]_{23} \\ [\boldsymbol{\Phi}^S_{\delta_t^m}]_{31} & [\boldsymbol{\Phi}^S_{\delta_t^m}]_{32} & [\boldsymbol{\Phi}^S_{\delta_t^m}]_{33} \end{bmatrix} = \begin{bmatrix} [\boldsymbol{\Phi}^S_{\delta_t^m}]_{11} & \boldsymbol{0} & [\boldsymbol{\Phi}^S_{\delta_t^m}]_{13} \\ [\boldsymbol{\Phi}^S_{\delta_t^m}]_{21} & \boldsymbol{0} & [\boldsymbol{\Phi}^S_{\delta_t^m}]_{23} \\ [\boldsymbol{\Phi}^S_{\delta_t^m}]_{31} & \boldsymbol{0} & [\boldsymbol{\Phi}^S_{\delta_t^m}]_{33} \end{bmatrix}, \tag{28}$$

with $S \in \{I, II\}$ and $m = 0, 1$. This highlights that the current state vector $\boldsymbol{s}_i^S$ is not dependent on the previous pressure value (*i.e.*, the pressure variable and its associated rows and columns in $\boldsymbol{\Phi}$ can be ignored without affecting $\boldsymbol{u}$ or $\boldsymbol{v}$.) Thus, we eliminate the rows and columns related to the pressure variable and proceed with the analysis by simultaneously diagonalizing the time stepping matrix,

$$\boldsymbol{T}^{-1} \begin{bmatrix} [\boldsymbol{\Phi}^S_{\delta_t^0}]_{11} & [\boldsymbol{\Phi}^S_{\delta_t^0}]_{13} \\ [\boldsymbol{\Phi}^S_{\delta_t^0}]_{31} & [\boldsymbol{\Phi}^S_{\delta_t^0}]_{33} \end{bmatrix} \boldsymbol{T} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots), \tag{29}$$

$$\boldsymbol{T}^{-1} \begin{bmatrix} [\boldsymbol{\Phi}^S_{\delta_t^1}]_{11} & [\boldsymbol{\Phi}^S_{\delta_t^1}]_{13} \\ [\boldsymbol{\Phi}^S_{\delta_t^1}]_{31} & [\boldsymbol{\Phi}^S_{\delta_t^1}]_{33} \end{bmatrix} \boldsymbol{T} = \mathrm{diag}(\gamma_1, \gamma_2, \ldots). \tag{30}$$

Then, the authors of [12] prove that, in the two-grid case with F- and FCF-relaxation, the global space-time error vector at the C-points can be reduced in the mass matrix norm by a convergence

DOI: 10.1002/nla

factor of at least,

$$c_f^F = \max_\omega \left\{ |\lambda_\omega^{c_0^1} - \gamma_\omega| \frac{1-|\gamma_\omega|^{N_t^1}}{1-|\gamma_\omega|} \right\}, \tag{31}$$

$$c_f^{FCF} = \max_\omega \left\{ |\lambda_\omega^{c_0^1} - \gamma_\omega| \frac{1-|\gamma_\omega|^{N_t^1-1}}{1-|\gamma_\omega|} |\lambda_\omega|^{c_0^1} \right\}, \tag{32}$$

respectively.

2.5.4. *Computation of MGRIT residual norm in experiment*

The standard Euclidean norm of the MGRIT residual in numerical experiments is computed using the coarse-grid residual $\boldsymbol{r}_\Delta$ (see Step 3 of Algorithm 1),

$$\begin{aligned} \|\boldsymbol{r}_\Delta\|_2^2 &= w_0 \|\boldsymbol{r}_\Delta^{\boldsymbol{v}}\|_2^2 + w_1 \|\boldsymbol{r}_\Delta^{p}\|_2^2 + w_2 \|\boldsymbol{r}_\Delta^{\boldsymbol{u}}\|_2^2 \\ &= w_0 (\boldsymbol{r}_\Delta^{\boldsymbol{v}})^T \boldsymbol{r}_\Delta^{\boldsymbol{v}} + w_1 (\boldsymbol{r}_\Delta^{p})^T \boldsymbol{r}_\Delta^{p} + w_2 (\boldsymbol{r}_\Delta^{\boldsymbol{u}})^T \boldsymbol{r}_\Delta^{\boldsymbol{u}} \end{aligned} \tag{33}$$

where $\boldsymbol{r}_\Delta^{\boldsymbol{v}}$, $\boldsymbol{r}_\Delta^{p}$ and $\boldsymbol{r}_\Delta^{\boldsymbol{u}}$ denote components in the residual vector corresponding to velocity, pressure and displacement. Further, the weights w_0, w_1 and w_3 are equal to 1 in the standard case.

However, the predicted decrease of the residual is measured in the mass matrix norm

$$\|\boldsymbol{r}_\Delta\|_{\mathfrak{M}}^2 = \|\boldsymbol{r}_\Delta^{\boldsymbol{v}}\|_{\mathfrak{M}^{\boldsymbol{v}}}^2 + \|\boldsymbol{r}_\Delta^{\boldsymbol{p}}\|_{\mathfrak{M}^{\boldsymbol{p}}}^2 + \|\boldsymbol{r}_\Delta^{\boldsymbol{u}}\|_{\mathfrak{M}^{\boldsymbol{u}}}^2, \tag{34}$$

where $\mathfrak{M}^{\boldsymbol{x}}$ is a block $N_t^1 \times N_t^1$ diagonal matrix with each block corresponding to a spatial mass matrix for variable $\boldsymbol{x}$ (see Section 2.5.3 and [12]). Thus, a modification of measuring the observed reduction of the residual norm per MGRIT iteration is advisable, and hence, proposed: instead of computing the computationally more demanding mass matrix norm and the less accurate standard Euclidean norm, the weights $w_0 = 1$, $w_1 = 0$ and $w_2 = 1/c_0^1$ are chosen to improve the measurement of experimental convergence factors. This yields

$$(\boldsymbol{r}_\Delta^{\boldsymbol{v}})^T \mathfrak{M}^{\boldsymbol{v}} \boldsymbol{r}_\Delta^{\boldsymbol{v}} \approx (\boldsymbol{r}_\Delta^{\boldsymbol{v}})^T \boldsymbol{I} \boldsymbol{r}_\Delta^{\boldsymbol{v}}, \tag{35}$$

$$(\boldsymbol{r}_\Delta^{\boldsymbol{u}})^T \mathfrak{M}^{\boldsymbol{u}} \boldsymbol{r}_\Delta^{\boldsymbol{u}} \approx (\boldsymbol{r}_\Delta^{\boldsymbol{u}})^T \left(\frac{1}{c_0^1} \boldsymbol{I} \right) \boldsymbol{r}_\Delta^{\boldsymbol{u}}, \tag{36}$$

Thus, the approximate value of the mass matrix norm of the residual at the C-points is given as,

$$\|\boldsymbol{r}_\Delta\|_{\mathfrak{M}} \approx \left((\boldsymbol{r}_\Delta^{\boldsymbol{v}})^T \boldsymbol{r}_\Delta^{\boldsymbol{v}} + \frac{1}{c_0^1} (\boldsymbol{r}_\Delta^{\boldsymbol{u}})^T \boldsymbol{r}_\Delta^{\boldsymbol{u}} \right)^{1/2}. \tag{37}$$

It is important to note that this proposition only changes how the solution progress (*i.e.*, reduction of residual norm) is measured. But it does neither affect the coarse-grid update nor change the numerical solution.

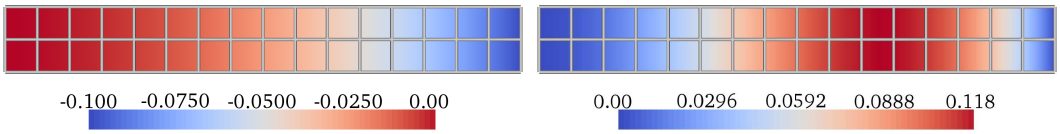
2.6. *Implementation details*

For the numerical experiments, the finite element software tool CHeart [13] was employed. In CHeart, Scheme I was available as the default scheme for linear elasticity, whereas Scheme II was implemented as part of this work. Wrapper routines were written to incorporate the MGRIT algorithm into CHeart using the open-source library XBraid [22], a non-intrusive implementation of the MGRIT algorithm.

Here, we introduced separate MPI groups and communicators in space and time to maintain the capability of CHeart to parallelize in the spatial domain by using domain decomposition methods while enabling independent parallelization in the temporal domain. That is, one can parallelize in space, time or in space-time.

Moreover, the XBraid option to skip work on the first down-cycle is used. Note, due to the linearity of the problem, the operator $\boldsymbol{\Phi}_{\delta_t^m}^S$ is only computed once for each time step size (*i.e.*, time grid level). This significantly reduces computational work compared to re-computing the operator for each time step.

DOI: 10.1002/nla



(a) Initial velocity in x-direction. (b) Initial velocity in y-direction.

Figure 2. Discretization using 16 x 2 quadrilateral elements and initial velocity.

3. RESULTS

In the following, the methods described in Section 2 are used for a linear beam problem. For this purpose, consider the domain $\Omega(0) = [0, 8] \times [0, 1]$, final time $T = 1024$ and material parameters $\mu = \rho = 1$. The spatial discretization uses 16 x 2 quadrilateral elements (mesh size $\delta_x = \delta_y = 0.5$) and initial conditions are given as $\hat{\boldsymbol{v}}_0 = [-x^2/640, x^2 \cdot (8 - x)/640]^T$, see Figure 2. The initial condition is also used as initial guess at all time steps for MGRIT. Further, no displacement boundary conditions are prescribed at $\Gamma^D = \Omega|_{x=0}$.

We consider fine grid time step sizes $\delta_{N_t^0} \in \{1, 0.1, 0.01, 0.001\}$, coarsening factors $c_0^1 \in \{2, 4, 8, 16, 32\}$, and $c_{m-1}^m = 2$ for $2 \leq m \leq m_l$. The stopping criterion on the residual norm is selected as $\|\boldsymbol{r}_\Delta\|_{\mathfrak{M}} \leq 5 \cdot 10^{-9}/\sqrt{\delta_x \delta_y \delta_{N_t^0}}$ (see Equation 37), with the maximum number of MGRIT iterations set to 60 iterations. Unless noted otherwise, reported experimental convergence factors are the global maximum values.

3.1. *Numerical solution of beam oscillation*

The initial velocity distribution over the cantilever beam length (see Figure 2) causes the free end to first deflect in the negative x- and positive y-directions, see Figure 3. The elastic stresses cause the beam to decelerate and move downward, passing its initial position and deflecting in negative y-direction. The beam deformation follows a repeatable deflection pattern.

3.2. *Effect of time step size on amplitude of oscillation*

Backward Euler time integration schemes introduce artificial numerical diffusion. Its effect on the sum of kinematic and potential energy of the system over time depends on the time step size, where we expect energy conservation in the asymptotic limit $\delta_{N_t^0} \to 0$. The artificial damping of the system causes the amplitude of oscillation to become smaller over time. The backward Euler time integration scheme exhibits quick damping for $\delta_{N_t^0} = 1$ irrespectively of the chosen scheme (Scheme I and Scheme II). The effect of numerical damping becomes smaller for smaller $\delta_{N_t^0}$, where we note that both schemes reproduce the amplitudes of the beam oscillation with comparable quality for $\delta_{N_t^0} = 0.001$, see Figure 3. Further, considering a second-order symplectic Verlet scheme (which maintains the amplitude of oscillation) with $\delta_{N_t^0} = 0.001$ as reference, we observe how both schemes tend toward the same solution with $O(\delta_{N_t^0})$, see Figure 4, such that the mismatch between both schemes becomes negligible (which is expected to be the case for time step sizes in practical applications).

It is important to note that MGRIT converges to the same solution (within the selected solver tolerance) as sequential time stepping on the fine grid. Thus, the converged numerical solution obtained with the MGRIT algorithm suffers from the same amount of numerical damping as the numerical solution from sequential time stepping on the fine grid.

3.3. *Convergence in the two-grid case*

3.3.1. *MGRIT with Scheme I*

If Scheme I is employed as a one-step integrator on the fine and coarse grid (that is, $\boldsymbol{\Phi}_{\delta_t^0}^I$ and $\boldsymbol{\Phi}_{\delta_t^1}^I$) in a two-level algorithm, we observe divergence in the numerical experiments for all considered test cases.

DOI: 10.1002/nla

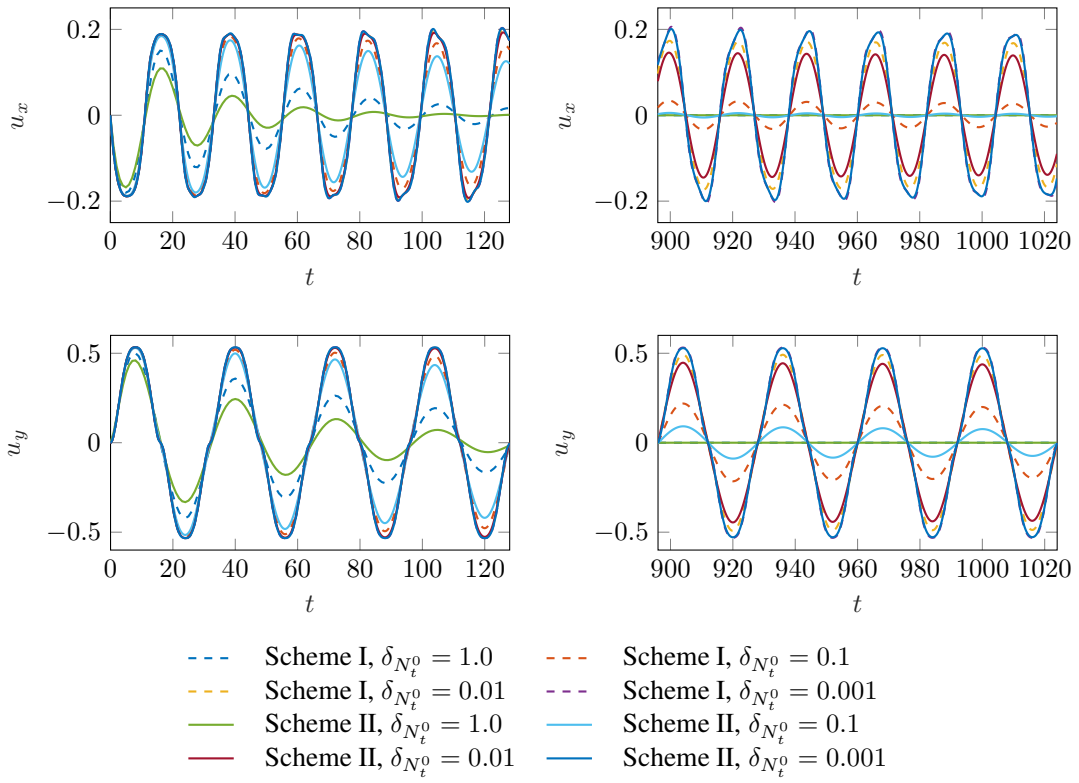


Figure 3. Displacement $\boldsymbol{u}$ of the tip of the cantilever beam (initial position $[8, 0.5]^T$) in the x- and y-directions for $t \in [0, 128]$ and $[896, 1024]$ for Scheme I and Scheme II and time step sizes $\delta_{N_t^0} \in \{1, 0.1, 0.01, 0.001\}$. Note that numerical damping is reduced with $\delta_{N_t^0} \to 0$.

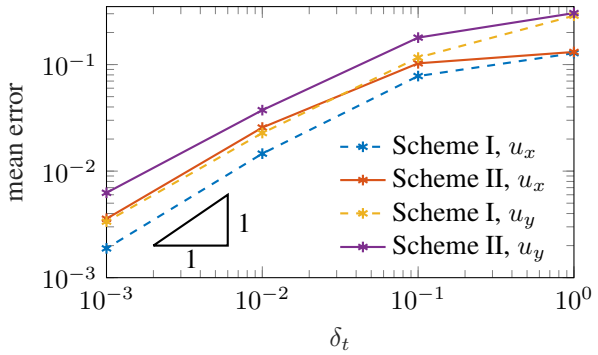


Figure 4. Mean absolute error of the displacement of the beam's tip for $t \in [0, 1024]$. Here, a second-order symplectic Verlet scheme with $\delta_{N_t^0} = 0.001$ was used as reference.

To qualitatively investigate what leads to the divergence of MGRIT in conjunction with Scheme I, we track the current position of the tip of the cantilever beam (initial coordinate $[8, 0.5]^T$ at $t = 0$) over time with the time horizon $t \in [0, 64]$. The time step size is $\delta_{N_t^0} = 1$ with coarsening factor $c_0^1 = 2$. FCF-relaxation is employed. Figure 5 shows the current approximation of the tip's displacement with respect to the initial position over time for a number of algorithmic steps, for example, after FCF-relaxation, after restriction, after the coarse-grid solve, *etc.* (where we extract the values on return from applying $\boldsymbol{\Phi}_{\delta_t^0}^I$ and $\boldsymbol{\Phi}_{\delta_t^1}^I$, respectively). The data in Figure 5 highlight how the current approximation of the cantilever's tip first improves. Though, already during the first MGRIT iteration an instability is introduced by the coarse-grid update which is then amplified in

DOI: 10.1002/nla

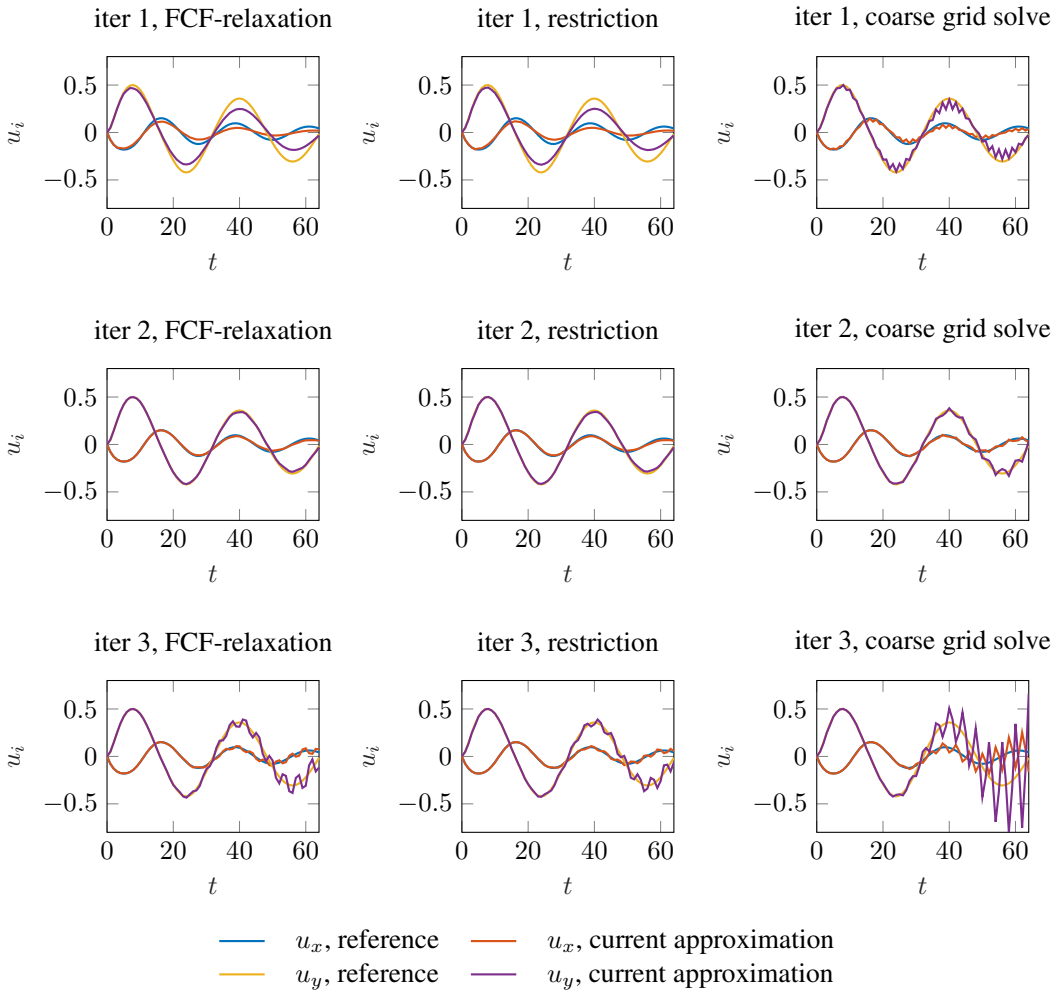


Figure 5. Current approximation of the tip displacement for two-grid MGRIT using Scheme I with $c_0^1 = 2$, $\delta_{N_t^0} = 1$, $t \in [0, 64]$ and FCF-relaxation compared with a reference solution from sequential time stepping. Note, how the coarse grid update introduces an instability which is amplified in subsequent steps.

subsequent steps. The observed phenomena are neither physical nor part of the mathematical model, however, they are in line with observations in the literature, see [9, 10]. Previously this has been a limiting factor for parallel-in-time integration and the dynamic elasticity equation.

On the other hand, the divergence of the numerical algorithm is reflected by the very large theoretical convergence bounds, *i.e.*, $c_f^F, c_f^{FCF} >> 1$, confirming experimental observations.

3.3.2. MGRIT with Scheme II

Using Scheme II as the fine- and coarse-grid integrators (that is, $\mathbf{\Phi}_{\delta_t^0}^{II}$ and $\mathbf{\Phi}_{\delta_t^1}^{II}$) in a two-level algorithm, we observe worst-case convergence factors of smaller than 1 (*i.e.*, residual norm is decreased for all iterations) for a range of different coarsening factors c_0^1 and for all considered time step sizes, see Figure 6.

We note that for $\delta_{N_t^0} = 1$, both experimental and predicted convergence factors are in excellent agreement and that the predicted values of c_f^F and c_f^{FCF} are a sharp upper bound despite the approximation of the computed residual norm, see Section 2.5.4. Here, the convergence factors first increase as the coarsening factor c_0^1 increases, before decreasing due to the small coarse-grid size $N_t^{m_l-1}$.

DOI: 10.1002/nla

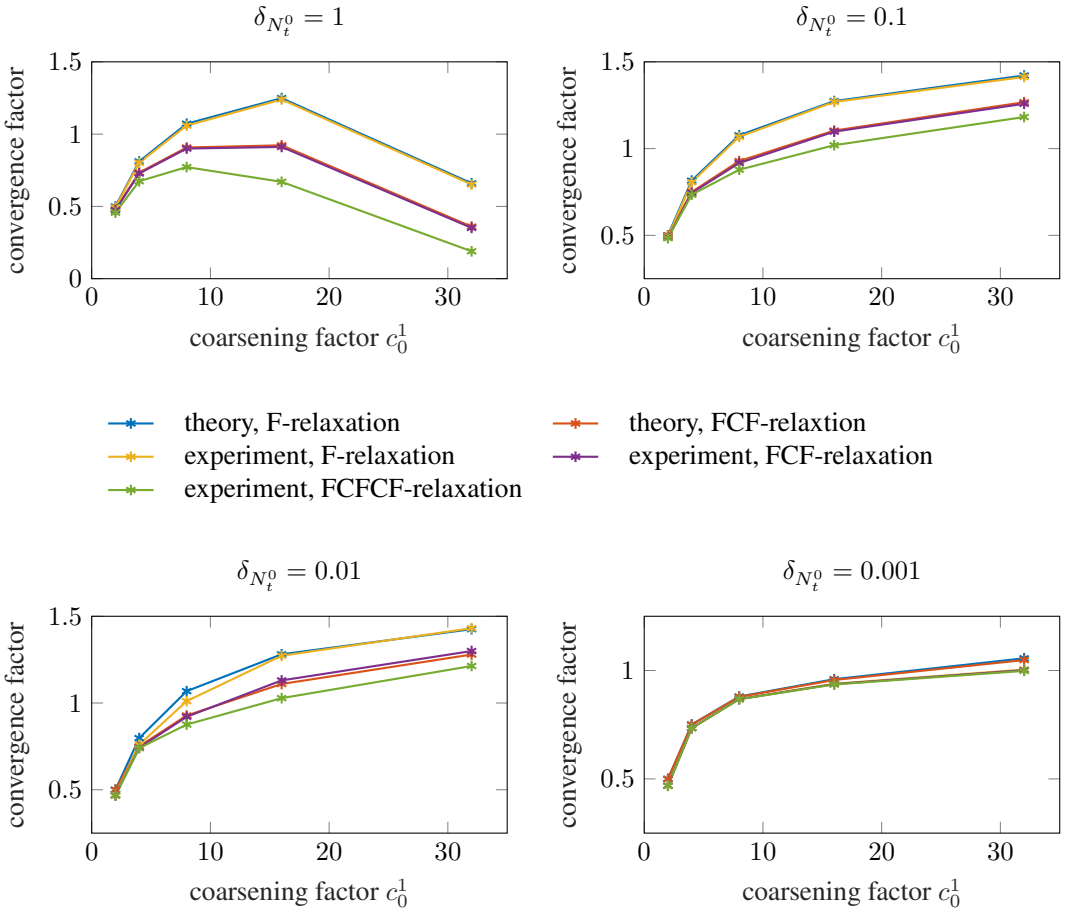


Figure 6. Predicted and measured convergence factor for a two-grid algorithm with various fine grid time step sizes $\delta_{N_t^0} \in \{1.0, 0.1, 0.01, 0.001\}$ and coarsening factors $c_0^1 \in \{2, 4, 8, 16, 32\}$.

On the other hand, for $\delta_{N_t^0} \in \{0.1, 0.01, 0.001\}$ we observe an increase in predicted and observed convergence factors as the coarsening factor c_0^1 increases. Again, predicted and observed convergence factors are in excellent agreement for almost all considered cases. Only for $\delta_{N_t^0} = 0.01$ is the maximum observed convergence factor larger than the predicted upper bound. The observed rate is 1% larger for both F-relaxation with $c_0^1 = 32$ and FCF-relaxation with $c_0^1 \in \{16, 32\}$. Again, this is likely due to the approximate residual norm computation.

Further, we note that additional relaxation steps can be beneficial for large fine grid time step sizes, whereas the effect is negligible for $\delta_{N_t^0} = 0.001$, thus suggesting that relaxation can be omitted for small fine grid time step sizes to reduce computational work without sacrificing convergence.

Finally, we note that in all considered cases we do not observe any instability as described in Section 3.3.1 and the previous work [9, 10]. For example, Figure 7 illustrates the position of the tip of the cantilever after the first three MGRIT iterations with $t \in [0, 64]$, $\delta_{N_t^0} = 0.1$, and $c_0^1 = 2$. Here, no artificial amplification of the amplitude of oscillation is observed, in contrast to the case of MGRIT with Scheme I, see Figure 5.

The results in this section highlight the benefit of using theoretical upper bounds, as given in [12], as a tool to estimate experimental convergence *a priori* and to design convergent MGRIT algorithms with guaranteed worst-case convergence factors.

DOI: 10.1002/nla

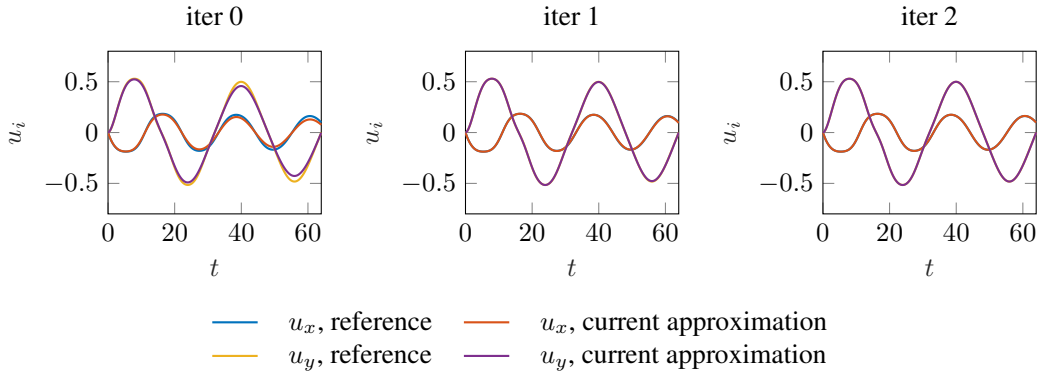


Figure 7. Current approximation of the tip displacement for two-grid MGRIT using Scheme II with coarsening factor $c_0^1 = 2$, fine grid time step size $\delta_{N_t^0} = 0.1$, $t \in [0, 64]$ and FCF-relaxation compared with reference solution from sequential time stepping. No instability is observed with Scheme II in contrast to using Scheme I, see Figure 5.

3.4. Convergence in the multigrid case

In the two-grid case, only MGRIT with Scheme II yielded a convergent algorithm. Thus, we neglected MGRIT with Scheme I in the following investigation of the convergence in the multigrid case and focus solely on MGRIT with Scheme II.

3.4.1. MGRIT with Scheme II

Here, we employ multilevel hierarchies with $m_l \in \{3, 4, 5\}$ for $\delta_{N_t^0} \in \{1, 0.1, 0.01\}$ and $m_l \in \{3, 4, 5, 6\}$ for $\delta_{N_t^0} = 0.001$. Further, we consider combined coarsening factors of $c_0^{m_l-1} \in \{4, 8, 16, 32, 64\}$, where we only vary c_0^1 but select $c_{m-1}^m = 2$ for $m = 2, \ldots, m_l - 1$. The MGRIT algorithm is started with a forward solve on the coarsest time grid (*i.e.* use of the XBraid skip-first-down option), performs an initial V-cycle and full multigrid (FMG) cycles for all following MGRIT iterations. Scheme II is employed as one-step integrator on all grid levels and one V-cycle is performed as post-relaxation step at each FMG level. To provide a better indicator for overall performance of the MGRIT algorithm, we report the mean of the experimental convergence factor over all MGRIT iterations.

Figure 8 reports mean experimental convergence factors over the combined coarsening factor $c_0^{m_l-1} = \prod_{m=0}^{m=m_l-2} c_m^{m+1}$. As illustrated, the best convergence factor for a particular combined coarsening factor is available through the use of FMG-cycles with more levels and slower coarsening between levels, as opposed to FMG-cycles with more aggressive coarsening between levels. Generally, observed convergence factors are significantly smaller than in the two-grid case. Thus, the use of FMG-cycles allows more aggressive coarsening and yields more potential for parallelism in the temporal domain.

If $\delta_{N_t^0}$ and $c_0^{m_l-1}$ are kept fixed, the mean convergence factor improves with growing m_l. For example, for $\delta_{N_t^0} = 0.1$ and $c_0^{m_l-1} = 64$ the mean convergence factor is approximately 0.92 for $m_l = 5$ but 1.07 for $m_l = 4$ and 1.22 for $m_l = 3$[‡]. Thus, one can obtain a moderately convergent instead of a slowly divergent algorithm simply by introducing an additional intermediate time grid level but with the same fine and coarsest grid size.

3.4.2. Timing results

In this section, we present speedup results for an MGRIT algorithm that employs Scheme II and

[‡]In particular, a measured mean convergence factor of larger than 1 indicates that the residual for the considered problem and a given algorithm cannot satisfy its convergence criteria within the performed 60 iterations.

DOI: 10.1002/nla

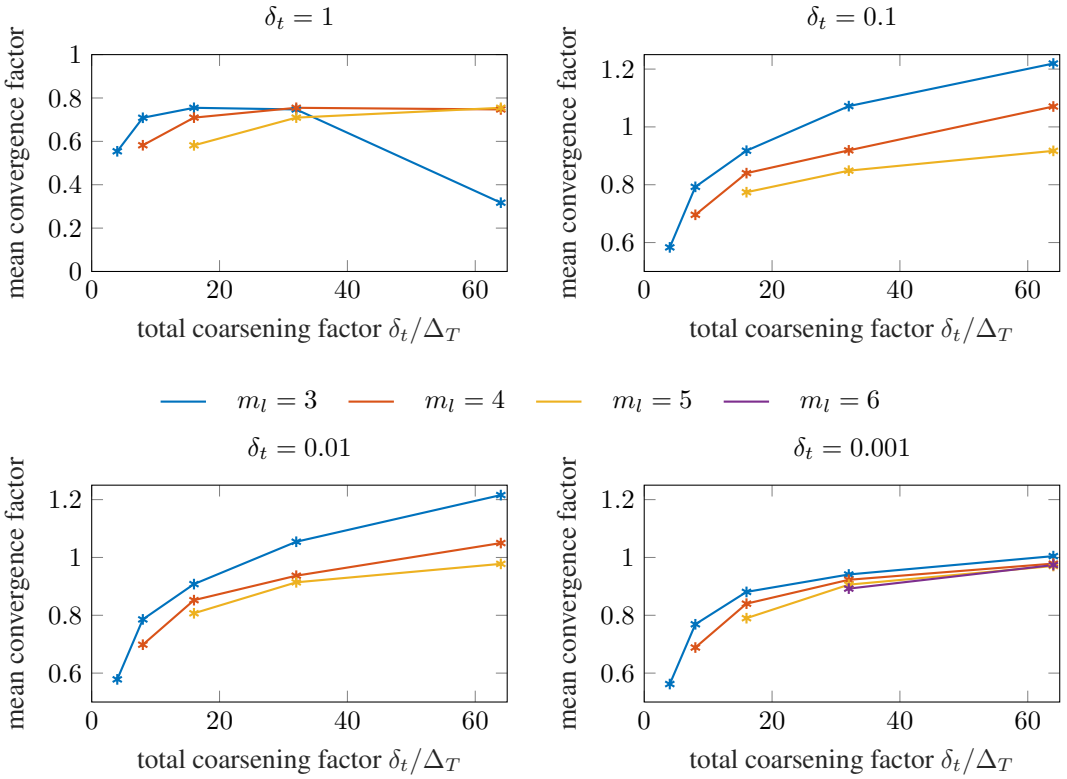


Figure 8. Measured mean convergence factor for m_l-grid MGRIT algorithm with F-cycles and Scheme II for fine grid time step sizes $\delta_{N_t^0} \in \{1, 0.1, 0.01, 0.001\}$. Note, that the x-axis corresponds to the combined coarsening factor $c_0^{m_l-1}$.

$m_l \in \{4, 5, 6\}$ time grid levels with a coarsening factor of two between all time grid levels. Here, we also investigate the effect of using up to four V-cycles as post-relaxation at each FMG-level. The time step size is selected as $\delta_{N_t^0} = 0.0005$ with $T = 64$. The Euclidean norm with $w_0 = 1$, $w_1 = 0$, $w_2 = 1/c_0^1$ (see Equation 33) is used to measure solution progress. To investigate the dependency of the wall clock time for MGRIT on the convergence criterion, we employ *tight*, *medium* and *loose* tolerances of $4.472 \cdot 10^{-7}$, $4.472 \cdot 10^{-6}$ and $4.472 \cdot 10^{-5}$.

All reported timing results were obtained on ASES (Intel Xeon E5-2680 v2 architecture, 20 cores @2.8GHz and 256GB RAM per node) at University of Stuttgart, Germany, with $\{16, 32, 48, \ldots, 192\}$ processors using 16 processors per compute node[§]. Data export was switched off and status messages were restricted to a bare minimum to ensure that reported wall clock times are almost entirely dominated by computation and communication costs but not data I/O tasks. The elapsed wall clock time for the sequential time stepping algorithm (using Scheme II) was 263 seconds, which is established as baseline.

Firstly, we note that the MGRIT solution with a loose tolerance of $4.472 \cdot 10^{-5}$ is a good approximation of the sequential time stepping solution. For example, Figure 9 illustrates the position of the cantilever's tip (initial position $[8, 0.5]^T$) and its velocity over time. Here, one can appreciate that all fine scale details in the solution are governed by the sequential time stepping solution as well as the MGRIT solution.

As Figure 10 illustrates, a 4-level MGRIT solver converges to its tolerance in less iterations than a 6-level MGRIT solver for a given number of V-cycles per FMG-level. On the other hand,

[§]Employing one compute node, a preliminary experiment showed best performance of our implementation at 16 processors per node.

DOI: 10.1002/nla

# V-cycles	$m_l = 4$	$m_l = 5$	$m_l = 6$
1	0.92	0.95	0.82
2	1.08	1.13	1.00
3	1.25	1.25	1.01
4	1.33	1.28	1.10

(a) MGRIT tolerance $tol = 4.472 \cdot 10^{-7}$.

# V-cycles	$m_l = 4$	$m_l = 5$	$m_l = 6$
1	1.43	1.38	1.02
2	1.70	1.80	1.33
3	1.87	1.80	1.37
4	2.12	1.77	1.46

(b) MGRIT tolerance $tol = 4.472 \cdot 10^{-6}$.

# V-cycles	$m_l = 4$	$m_l = 5$	$m_l = 6$
1	3.20	3.45	2.10
2	3.96	3.55	2.60
3	3.74	4.01	2.70
4	5.25	4.72	2.91

(c) MGRIT tolerance $tol = 4.472 \cdot 10^{-5}$.

# FC-relaxations	$m_l = 4$	$m_l = 5$	$m_l = 6$
0	1.95	1.28	0.61
1	1.83	1.13	0.54

(d) MGRIT tolerance $tol = 4.472 \cdot 10^{-5}$.

Table I. Measured speedup using 192 processors with $m_l \in \{4, 5, 6\}$. Table Ia - Table Ic with FMG-cycles and $\{1, 2, 3, 4\}$ V-cycles at each FMG-level; Table Id with V-cycles and F- or FCF-relaxation.

performing more V-cycles per FMG-level improves convergence factors significantly. Thus, the required number of iterations to solve the problem to solver tolerance decreases. With $m_l = 6$ levels, for example, MGRIT takes 37 iterations to satisfy the tight tolerance when using one V-cycle per FMG-level, however, it takes 25, 20 or 17 iterations when performing two, three or four V-cycles per FMG-level. We further note, that the residual norm is always decreased in subsequent iterations.

As observed in the previous paragraph, the required number of iterations drops when performing additional V-cycles at each FMG-level. Despite the additional *per-iteration* cost; however, the measured wall clock time of the algorithm decreases due to the significantly smaller convergence factors. Figure 11 - Figure 13 highlight this behavior, where each additional V-cycle per FMG-level yields a reduction in wall clock time for all $m_l \in \{4, 5, 6\}$. Here, the best speedups are observed for a 4-level MGRIT solver with four V-cycles per FMG-level. The measured speedup factor is 5.25, 2.12 and 1.33 for the three considered solver tolerances, see Table Ia - Table Ic. Further, the speedup is 5.25, 4.72 or 2.91 for the loose solver tolerance and a 4-, 5- or 6-level MGRIT solver and four V-cycles per FMG-level. Note, that the slope of the curves in Figure 11 and Figure 13 show better scaling of a 6-level MGRIT solver compared to a 4-level MGRIT solver due to the smaller coarse grid size and larger potential for parallelism. At processor counts of up to 192, no crossover point can be reached when performing only one V-cycle per FMG-level and imposing the tight tolerance. For all other combinations of tolerances and cycling strategies considered in this section, however, a crossover point is reached.

Lastly, we compare the performance of the FMG-cycle algorithm to a V-cycle algorithm with F- or FCF-relaxation. At the loose solver tolerance, a speedup of 1.95 is achieved using 192 processors with $m_l = 4$ and F-relaxation, see Table Id. Additional FC-relaxation steps, however, result in a more expensive V-cycle algorithm. No speedups are observed for the medium and tight tolerances (not included in Table I). Further, we note the benefit of using FMG-cycles with additional V-cycles per FMG-level over a V-cycle algorithm. For example, for $m_l = 4$ the FMG-cycle algorithm can solve the problem to a tighter solver tolerance compared to the V-cycle algorithm with a similar speedup. The best measured speedups are 5.25 for a FMG-cycle algorithm and 1.95 for a V-cycle algorithm.

4. DISCUSSION

In the previous sections, two different backward Euler time discretization schemes were presented and investigated. Scheme I was considered as the default scheme in our application code CHeart [13], whereas Scheme II was proposed as an improvement for parallel-in-time methods.

DOI: 10.1002/nla

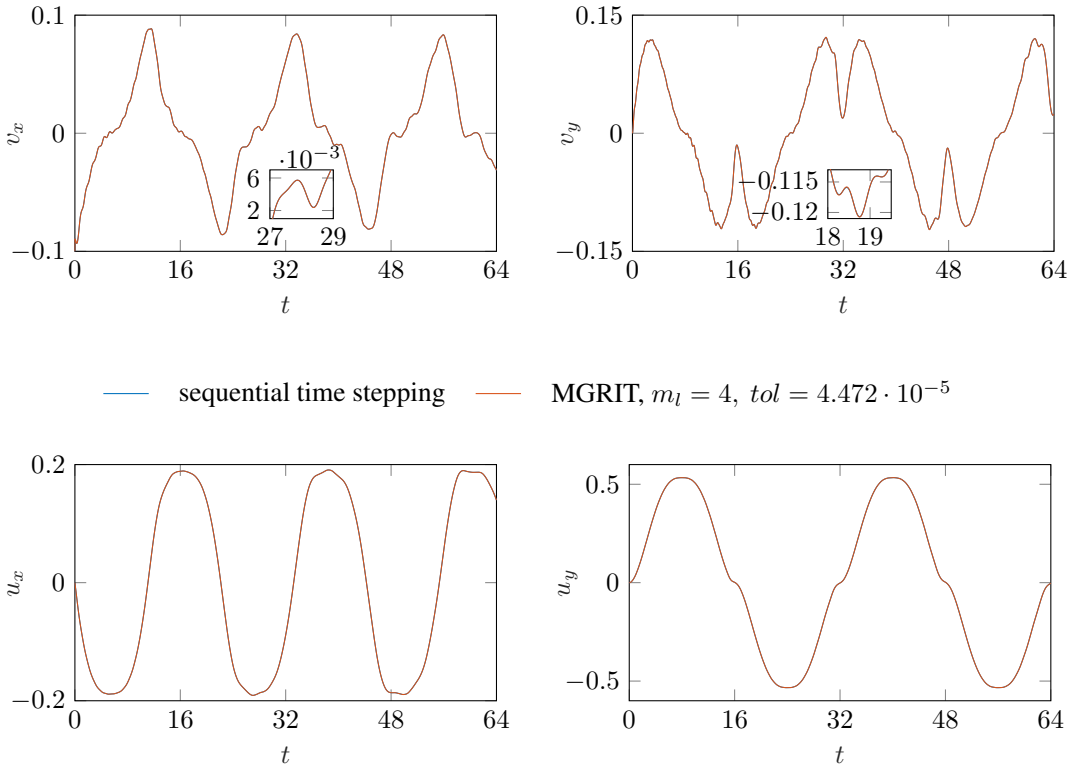


Figure 9. Comparison of velocity and displacement at the tip of the cantilever beam (initial position $[8, 0.5]^T$) for sequential time stepping and MGRIT with Scheme II ($m_l = 4$, $c^m_{m-1} = 2$ for all m), time step size $\delta_{N_t^0} = 0.0005$ and MGRIT tolerance $tol = 4.472 \cdot 10^{-5}$. Note, how fine time scale variations are resolved despite the relatively loose MGRIT convergence tolerance.

It was shown, that MGRIT with Scheme I exhibits strong instabilities for our parallel-in-time approach. The observations are in line with previous work in this field [9, 10]. Although MGRIT with Scheme II uses only a slightly different time discretization, a convergent scheme can be derived for a range of coarsening factors, both for two-grid and multilevel algorithms. Scheme II not only results in a stable algorithm when used with MGRIT but also yields a speedup over sequential time stepping while maintaining the accuracy of Scheme I for practical time step sizes.

Intuitively, the change from Scheme I to Scheme II changes how the discretized version of the stress tensor in Equation 6 is evaluated (compare Equation 12 and Equation 15), such that it is evaluated at given discrete time points across all levels in the time grid hierarchy in a consistent manner, see Section 2.5. Here, we note that both time discretizations are consistent and converge to the same numerical solution for decreasing time step sizes. By rewriting the time-discrete Equation 14,

$$\boldsymbol{u}_i = \boldsymbol{u}_{i-1} + \delta_{N_t} \boldsymbol{v}_i \tag{38}$$

$$= \boldsymbol{u}_{i-1} + \frac{\delta_{N_t}}{2}(\boldsymbol{v}_i + \boldsymbol{v}_{i-1}) + \frac{\delta^2_{N_t}}{2}(\frac{\boldsymbol{v}_i - \boldsymbol{v}_{i-1}}{\delta_{N_t}}), \tag{39}$$

one may argue that, for δ_{N_t}, Equation 39 becomes increasingly similar to the time-discrete Equation 11. Further, rewriting Equation 11,

$$\boldsymbol{u}_i = \boldsymbol{u}_{i-1} + \frac{\delta_{N_t}}{2}(\boldsymbol{v}_i + \boldsymbol{v}_{i-1}) = \boldsymbol{u}_{i-1} + \delta_{N_t}\boldsymbol{v}_{i-1} - \frac{\delta_{N_t}}{2}\boldsymbol{v}_{i-1} + \frac{\delta_{N_t}}{2}\boldsymbol{v}_i \tag{40}$$

$$= \boldsymbol{u}_{i-1} + \delta_{N_t}\boldsymbol{v}_{i-1} + \frac{\delta^2_{N_t}}{2}(\frac{\boldsymbol{v}_i - \boldsymbol{v}_{i-1}}{\delta_{N_t}}), \tag{41}$$

DOI: 10.1002/nla

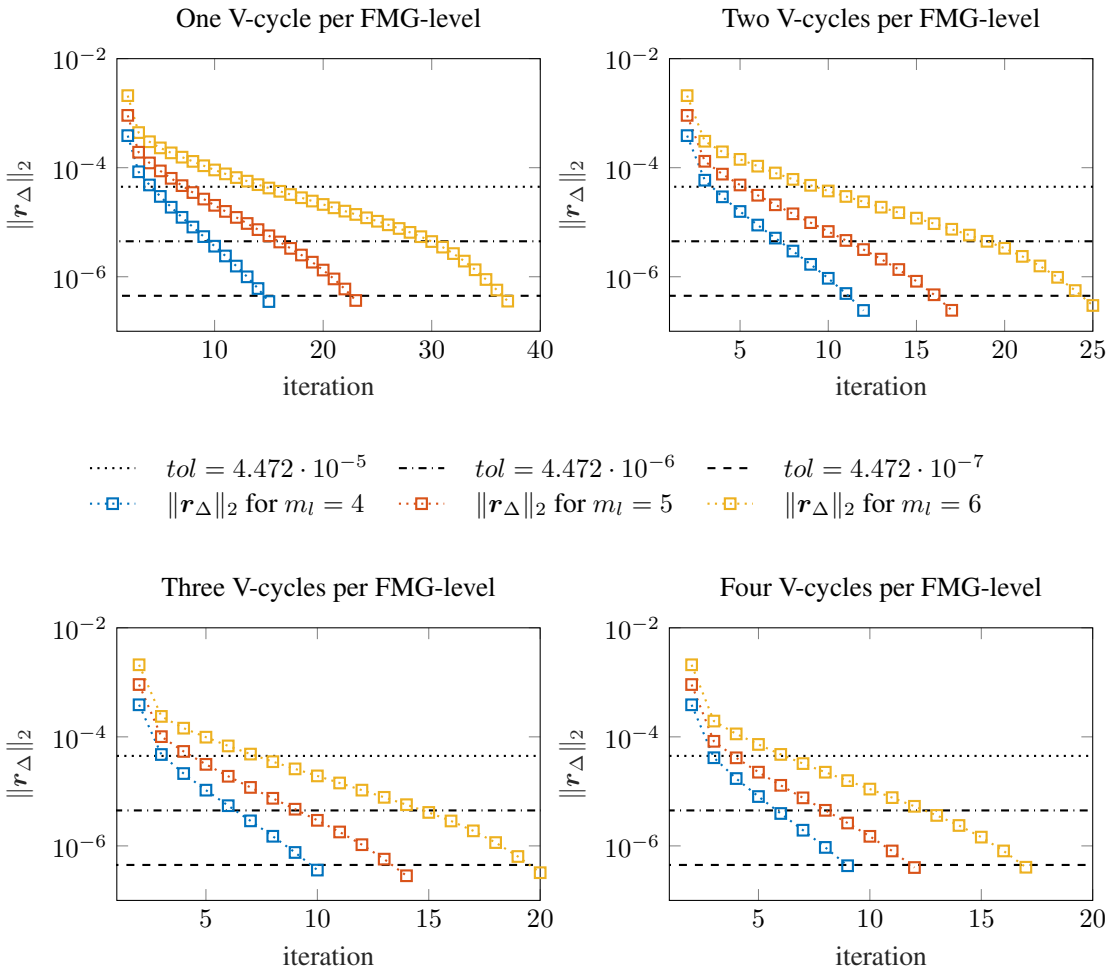


Figure 10. Decrease of MGRIT residual norm $\|r_\Delta\|_2$ (with $w_0 = 1$, $w_1 = 0$, $w_2 = 1/c_0^1$; see Equation 33) for $m_l \in \{4, 5, 6\}$ and $\{1, 2, 3, 4\}$ V-cycles per FMG-level. Note how the additional V-cycle per FMG-level yields a faster decrease in the residual norm. Thus, the required number of iterations to solve the problem to solver tolerance decreases. This causes a drop in wall clock time, as seen in Figure 11 - Figure 13, despite the higher per-cycle cost.

and assuming that the last term in Equation 41 can be neglected for small δ_{N_t}, one would effectively approximately discretize Equation 5 by an explicit step. This might be a hint for explaining the observed instability of MGRIT with Scheme I, however, numerical experiments with explicit time integration schemes were not considered in this work. Another possible explanation for the instability of MGRIT with Scheme I could be a potential requirement for the same discretization scheme for all coupled equations[¶]. On the other hand, the two-grid theory presented in [12], when used as an *a priori* tool for estimating convergence of MGRIT with Scheme I, provides a more mathematical and clear pathway for relating observed divergence in numerical experiments with the employed time-discretization.

In the case of using MGRIT with Scheme II, convergence was predicted for two-grid algorithms with F- and FCF-relaxation and a range of coarsening factors. The predictions matched quite closely with observations in the numerical experiments, supporting the use of the analysis presented in [12] as a powerful tool to design convergent algorithms *a priori*.

[¶]Scheme II uses a backward Euler discretization for all terms.

DOI: 10.1002/nla

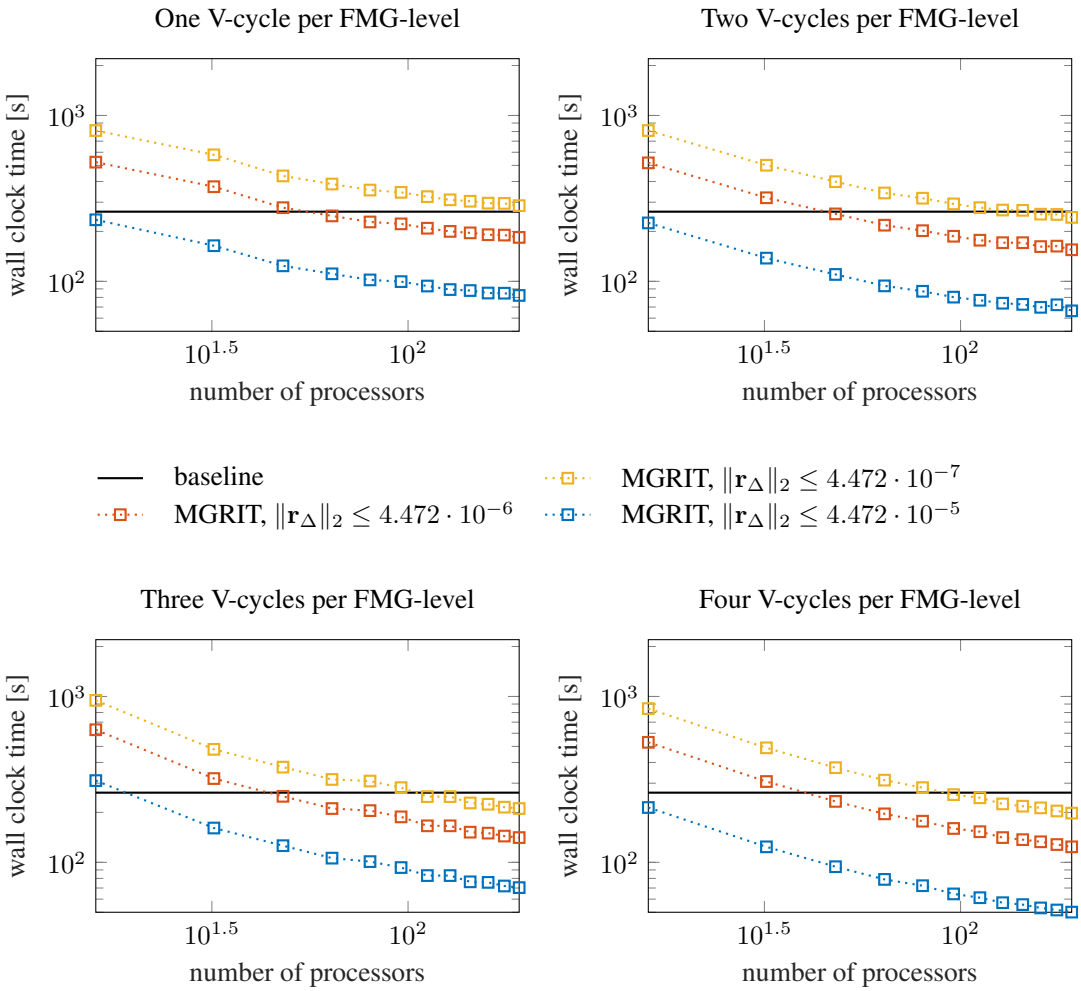


Figure 11. Wall clock time for MGRIT with Scheme II and FMG-cycles with $m_l = 4$ and $c^m_{m-1} = 2$ for all m.

Although the analysis presented in [12] is restricted to the two-grid case, the convergent and efficient multilevel algorithm was a straightforward generalization of its two-grid counterpart. Here, the use of FMG-cycles was beneficial to accelerate convergence in the true multilevel case, enabling larger combined coarsening factors $c_0^{m_l-1}$ compared to the two-grid case. It was also shown that, for a given combined coarsening factor $c_0^{m_l-1}$, slow temporal coarsening (and thus, more time grids) can improve convergence over faster temporal coarsening with less time grids. Furthermore, performing additional V-cycles at each FMG-level makes the coarse-grid solve more powerful and improves convergence significantly.

Further, timing results for the time-parallel algorithm were presented in Section 3.4.2. Using 192 processors, a speedup of 5.25 was achieved for a 4-level algorithm with four V-cycles per FMG-level. Similarly, speedups of 4.72 and 2.91 were shown for 5- and 6-level algorithms. It was also demonstrated, that the use of FMG-cycles resulted in a better speedup than the use of V-cycles with F- and FCF-relaxation. The reported wall clock times are almost entirely dominated by computation and communication, however, data export, for example, is a completely serial process for sequential time-stepping while it is parallel for the MGRIT algorithm. Thus, for practical applications that include data I/O tasks, larger speedups can be expected. Lastly, Figure 13 (for example, 6-level algorithm with four V-cycles per FMG-level) shows scaling of the wall clock time at fixed spatial

DOI: 10.1002/nla

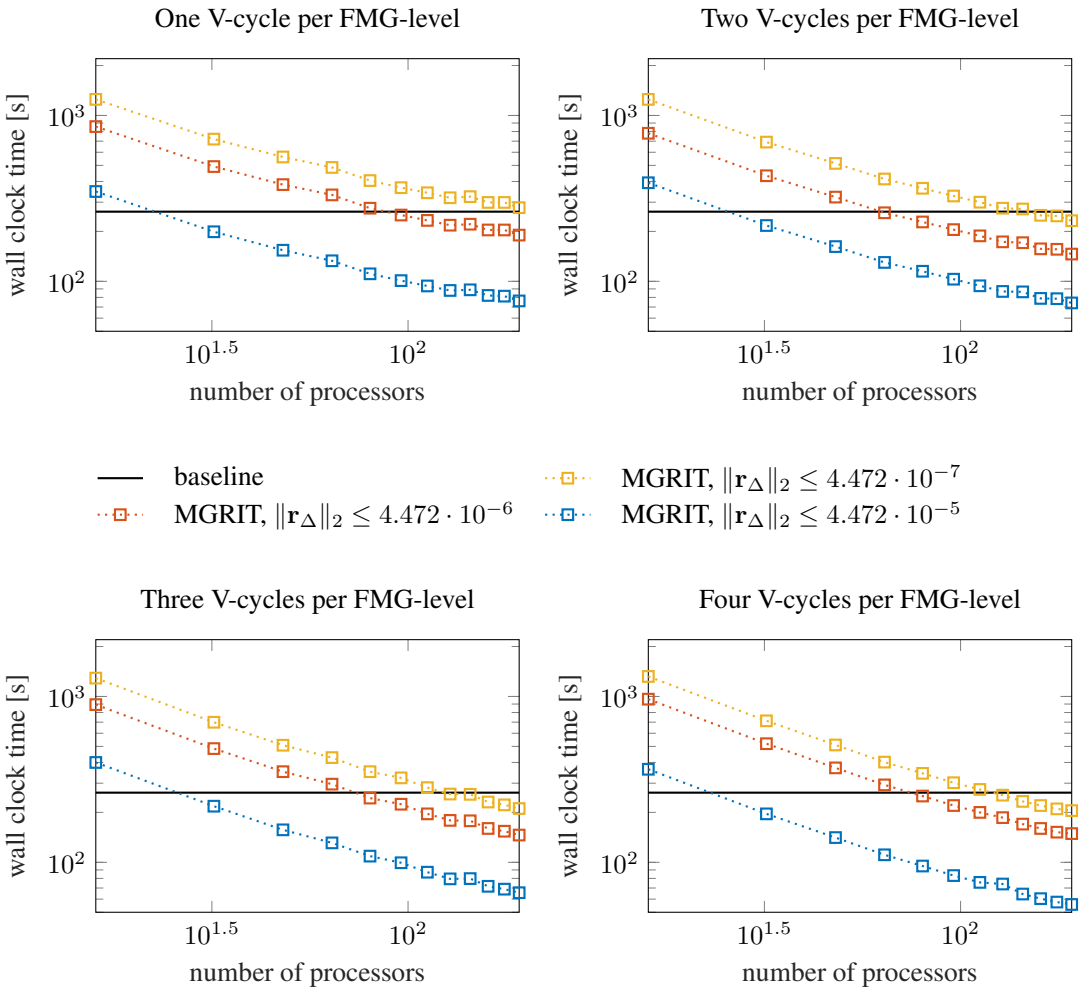


Figure 12. Wall clock time for MGRIT with Scheme II and FMG-cycles with $m_l = 5$ and $c^m_{m-1} = 2$ for all m.

problem size until 192 processors, which we expect to continue beyond the number of processors employed in this study.

In this work, we did not consider refinement of the spatial problem or spatial parallelism as the performance of MGRIT and the expected speedup are mainly dictated by the temporal dimension size. Spatial parallelism will degrade strong scaling, because communication overhead will be relatively larger. Larger spatial problems, however, will help to improve strong scaling because each time step will be more expensive, that is, communication overhead will be relatively smaller.

5. CONCLUSION

In this work, we provided an analysis of the convergence of the multigrid reduction in time algorithm for dynamic linear elasticity equations. It was shown that the two-grid convergence theory presented in [12] provides a mathematical explanation for the instability observed in previous work [9, 10]. Using the two-grid convergence theory as a design tool to estimate convergence of a backward Euler scheme (MGRIT with Scheme II) *a priori*, we were able to obtain a convergent parallel-in-time algorithm for a range of coarsening factors, advancing the application of parallel-in-time methods for second-order hyperbolic equations. In this study, the predicted theoretical convergence bounds

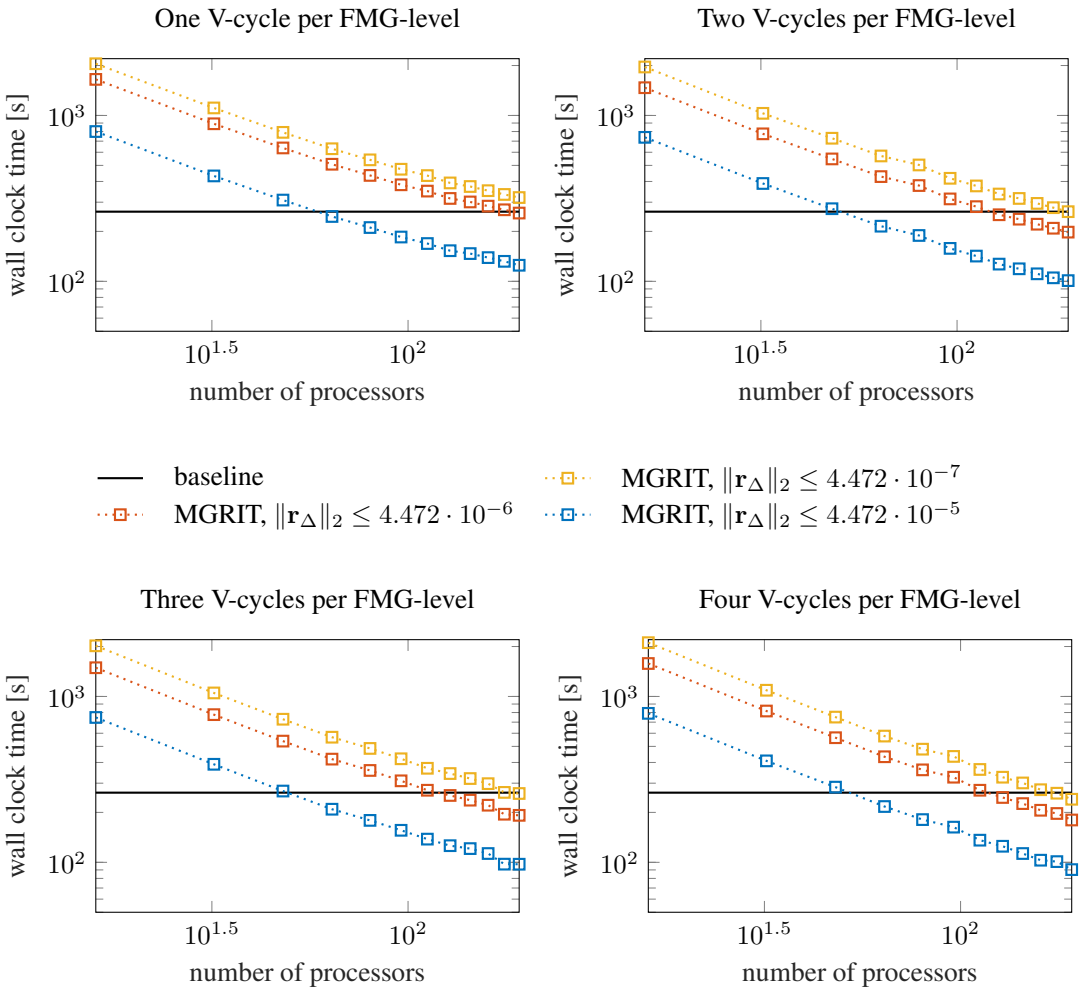


Figure 13. Wall clock time for MGRIT with Scheme II and FMG-cycles with $m_l = 6$ and $c^m_{m-1} = 2$ for all m.

were in excellent agreement with the worst-case convergence factors in numerical experiments. We then extended the algorithm to the multilevel case, noting that FMG-cycles and slow temporal coarsening can improve convergence compared to the two-grid algorithm. Further, performing additional V-cycles per FMG-level helped to improve the observed convergence factors significantly, yielding a reduction in the wall clock time. The best speedup achieved in this work was 5.25 (*i.e.*, a reduction of the wall clock time by approximately 81%), showing a competitive algorithm to sequential time stepping.

In future work, we will investigate other cycling strategies (*e.g.* switching from FMG- to V-cycles after a number of initial iterates) to improve and accelerate convergence and further reduce the wall clock time of the algorithm. Moreover, spatial coarsening will be investigated. Further, we aim to generalize the framework to efficiently solve the dynamic elasticity equation using a nonlinear hyper-elastic stress-strain relationship. Ultimately, we aim to extend the application of MGRIT to fluid-structure interaction problems, particularly in the field of biomedical engineering.

ACKNOWLEDGEMENT

DOI: 10.1002/nla

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007–2013)/ERC Grant Agreement No. 306757 (LEAD). D.N. would like to acknowledge funding from Engineering and Physical Sciences Research Council (EP/N011554/1 and EP/R003866/1).

REFERENCES

1. McCormick M, Nordsletten DA, Kay D, Smith NP. Simulating left ventricular fluid–solid mechanics through the cardiac cycle under LVAD support. J Comput Phys 2013; 244:80–96.
2. Hessenthaler A, Röhrle O, Nordsletten D. Validation of a non-conforming monolithic fluid-structure interaction method using phase-contrast MRI. Int J Numer Meth Bio 2016.
3. Bai Y, Sun D, Lin J, Kennedy D, Williams F. Numerical aerodynamic simulations of a NACA airfoil using CFD with block-iterative coupling and turbulence modelling. Int J Comput Fluid D 2012; 26(2):119–132.
4. Nievergelt J. Parallel methods for integrating ordinary differential equations. Commun ACM 1964/ 7(12):731–733.
5. Gander M. 50 years of time parallel time integration (In: Multiple Shooting and Time Domain Decomposition Methods). Springer 2015.
6. Garrido I, Espedal MS, Fladmark GE. A convergent algorithm for time parallelization applied to reservoir simulation (In: Domain Decomposition Methods in Science and Engineering). Springer 2005.
7. Samaddar D, Casper TA, Kim SH, Berry LA, Elwasif WR, Batchelor DB, Houlberg WA. Time parallelization of advanced operation scenario simulations of ITER plasma. J Phys Conf Ser 2013; 410(1):012032.
8. Schreiber M, Peixoto PS, Haut T, Wingate B. Beyond spatial scalability limitations with a massively parallel method for linear oscillatory problems. Int J High Perform C 2017.
9. Farhat C, Chandesris M. Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid-structure applications. Int J Numer Meth Eng 2003; 58(9):1397–1434.
10. Farhat C, Cortial CJ, Dastillung C, Bavestrello H. Time-parallel implicit integrators for the near-real-time prediction of linear structural dynamic responses. Int J Numer Meth Eng 2006; 67(5):697–724.
11. Cortial J, Farhat C. A time-parallel implicit method for accelerating the solution of non-linear structural dynamics problems. Int J Numer Meth Eng 2009; 77(4):451–470.
12. Dobrev VA, Kolev TZ, Petersson NA, Schroder JB. Two-level convergence theory for multigrid reduction in time (MGRIT). SIAM J Sci Comput 2017; 39(5):S501–S527.
13. Lee J, Cookson A, Roy I, Kerfoot E, Asner L, Vigueras G, Sochi T, Deparis S, Michler C, Smith NP, Nordsletten DA. Multiphysics Computational Modeling in CHeart. SIAM J Sci Comput 2016; 38(3):C150–C178.
14. Wriggers P. Nonlinear Finite Element Methods. Springer Berlin Heidelberg; 2008.
15. Simo JC, Wong KK. Unconditionally stable algorithms for rigid body dynamics that exactly preserve energy and momentum. Int J Numer Meth Eng 1991; 31(1):19–52.
16. Simo JC, Tarnow N. The discrete energy-momentum method. Conserving algorithms for nonlinear elastodynamics. Z Angew Math Phys 1992; 43(5):757–792.
17. Gander MJ, Vandewalle S. Analysis of the parareal time-parallel time-integration method. SIAM J Sci Comput 2007; 29(2):556–578.
18. Friedhoff S, MacLachlan S. A generalized predictive analysis tool for multigrid methods. Numer Linear Algebr 2015; 22(4):618–647.
19. Falgout RD, Manteuffel TA, O'Neill B, Schroder JB. Multigrid reduction in time for nonlinear parabolic problems: A case study. SIAM J Sci Comput 2017; 39(5):S298–S322.
20. Falgout RD, Friedhoff S, Kolev TZ, MacLachlan SP, Schroder JB. Parallel time integration with multigrid. SIAM J Sci Comput 2014; 36(6):C635–C661.
21. Trottenberg U, Oosterlee C, Schüller A. Multigrid. Academic Press, London, UK, 2001.
22. XBraid: Parallel multigrid in time. http://llnl.gov/casc/xbraid.